

**TRANSMITTAL OF APPEAL BRIEF (Large Entity)**Docket No.  
MCT-0126In Re Application Of: **Duane Le Allen**Serial No.  
08/925,703Filing Date  
September 9, 1997Examiner  
G. OpieGroup Art Unit  
2151Invention: **Method and Apparatus for Installing an Operating System**RECEIVED  
OCT 11 9 2001RECEIVED  
OCT 1 8 2001  
Group 2100TO THE ASSISTANT COMMISSIONER FOR PATENTS  
INTERFERENCE

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on August 29, 2001

The fee for filing this Appeal Brief is: **\$320.00**

- ☒ A check in the amount of the fee is enclosed.
- ☐ The Commissioner has already been authorized to charge fees in this application to a Deposit Account. A duplicate copy of this sheet is enclosed.
- ☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 20-1504  
A duplicate copy of this sheet is enclosed.

Signature

**Fred G. Pruner, Jr., Reg. No. 40,779**  
**Trop, Pruner & Hu, P.C.**  
**8554 Katy Freeway, Suite 100**  
**Houston, Texas 77002**  
**(713) 468-8880 [Phone]**  
**(713) 468-8883 [Fax]**

Dated: 10/02/01

I certify that this document and fee is being deposited on 10/02/01 with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

Signature of Person Mailing Correspondence

**Debra Cutrona**

Typed or Printed Name of Person Mailing Correspondence

CC:

#2 4/ App 2 9 (b) 1  
T.M.C. Beth Brown  
10/25/01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Duane Le Allen § Group Art Unit: 2151  
Serial No.: 08/925,703 §  
Filed: September 9, 1997 § Examiner: G. Opie  
For: METHOD AND APPARATUS § Atty. Dkt. No.: MICT-0126-US  
FOR INSTALLING AN § (MUEI-0305.00/US)  
OPERATING SYSTEM §

RECEIVED  
OCT 10 2001  
Group 2100

Board of Patent Appeals & Interferences  
Commissioner for Patents  
Washington, D.C. 20231

APPEAL BRIEF

Dear Sir:

Applicant hereby appeals from the Final Rejection dated June 15, 2001.

I. REAL PARTY IN INTEREST

The real party in interest is Micron Technology, Inc., the assignee of the present application by virtue of an assignment from Micron Electronics, Inc., the original assignee, by virtue of the original assignment recorded at Reel/Frame No. 9080/0067.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

The application was originally filed with claims 1-27. Claims 28-51 were added during prosecution of the application. Claims 1-33 have been canceled. Claims 34-51 have been finally rejected and are the subject of this appeal.

Date of Deposit: 10/02/01  
I hereby certify under 37 CFR 1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated above and is addressed to the Board of Patent Appeals & Interferences, Commissioner for Patents, Washington, DC 20231.  
Debra Cutrona

#### IV. STATUS OF AMENDMENTS

There are no unentered amendments.

#### V. SUMMARY OF THE INVENTION

Figure 1 illustrates a computer system 10 upon which the invention in one embodiment may be practiced. A personal computer 12 includes a monitor 14, a floppy disk drive 16, an optical disk drive 18, a hard disk drive 20, and a keyboard 22. The computer 12 illustrated is a desktop computer manufactured by Micron Electronics, Inc. of Nampa, Idaho. However, the invention is not so limited as computers by other manufacturers and other types of personal computers, such as laptop and hand-held computers, might also be employed. Specification, p. 7.

The computer 12 of Figure 1 also includes several add-on devices. More particularly, computer 12 is connected to a modem 24, a mouse 26, and a laser printer 28. The computer system 10 might, in other embodiments, omit such add-on devices or employ still other add-on devices, those illustrated merely being exemplary of add-on devices generally considered standard. There may be many variations on this theme. For instance, the modem 24 is shown internal to computer 12, but in other embodiments may be external. Also, the printer 28 may be a type of printer other than a laser printer and may even be some other form of input/output ("I/O") device such as a plotter or magnetic tape drive. Specification, pp. 7-8.

Figure 2 illustrates a method for installing an operating system on a computer such as computer 12. As mentioned above, an operating system must be installed for a user to interface with the computer. In turn, drivers must be installed for the computer 12 to communicate with each device. Thus, a computer is programmed with an operating system and drivers are typically installed with the operating system. Specification, p. 9.

The method of Figure 2 begins by generating an external configuration file containing information about a plurality of devices that might be associated with the computer 12. The plurality of devices will typically include those interfacing with the computer 12 through a variety of protocols such as industry standard architecture ("ISA"), peripheral component interconnect ("PCI"), small computer serial interface ("SCSI"), and integrated drive electronics ("IDE") protocols. Also, the plurality of devices may include all manner of types, such as monitors, sound cards, and modems from a variety of manufacturers. Although the external configuration file need not necessarily be so implemented, it may be conceptualized as a lengthy list of devices and associated information regarding their manufacturer, their type, and the protocol they employ. Specification, pp. 9-10.

As mentioned above, the OS installation software also contains a configuration file, which shall be referred to as an "internal" configuration file. The term "external" is used here to distinguish the configuration file of the invention from the internal configuration file of the OS installation software. The external configuration file is generated in much the same way as the internal configuration file as will be recognized by those in the art having the benefit of this disclosure. However, because the external configuration file is not part of the OS installation software, it can be frequently updated or modified as new devices are released or errors are identified in the internal configuration file. Specification, p. 10.

The method proceeds by next identifying a particular device associated with the computer, the particular device being one of the plurality of devices set forth in the external configuration file. This identification may be either automated or manual. If manual, the installer might be queried as to the identity of the various devices, which might introduce human error into the installation process. Because the external configuration file can be readily updated,

it is envisioned that a device not included in the external configuration file will rarely be encountered, although this is possible. Specification, pp. 10-11.

The method next proceeds by installing a driver for the device and the operating system on the computer according to the information stored in the external configuration file and the particular device's identity. This may be done either by replacing erroneous drivers installed by the OS installation software, replacing the erroneous drivers to be installed by the OS software so that the OS installation software will install the correct drivers, or by patching the drivers installed by the OS installation software to correct the errors they contain. Thus, driver installation may be performed prior to, contemporaneous with, or after installing the operating system. The embodiment of Figure 2 is finished once the OS is fully installed. However, in alternative embodiments, additional tasks may be performed if desired. For instance, various type of applications software might be loaded. Specification, p. 11.

Figures 3 and 4A-4C illustrate one particular variation of the embodiment in Figure 2. Referring now to Figure 3, this embodiment involves:

- (a) booting the computer 12 from a CD-ROM boot disk containing the software embodying the present invention;
- (b) selecting a type of installation;
- (c) selecting video, sound, modem, and controller types;
- (d) copying a WIN95 directory from a Windows® 95 compact disk to the hard drive 20 of the computer 12;
- (e) copying the files needed to install the drivers for each device as defined in the external configuration file OSI.INI;

(f) running the setup with a custom install file OSI.INF generated from the external configuration file OSI.INI; and

(g) modifying the registry for each device to correct the configuration information therein as defined in the external configuration file OSI.INI.

Figures 4A-4C and the discussion below flesh out this process. Specification, pp. 11-12.

In this particular embodiment, the computer 12 is a desktop personal computer such as is commercially available from Micron Electronics, Inc., of Nampa, Idaho, and the operating system is Windows® 95 available from Microsoft Corporation of Seattle, Washington. The instructions for performing the method in this particular embodiment are encoded on a removable, portable program storage medium such as floppy diskette 30 in Figure 1 inserted into the floppy disk drive 16 or a CD-ROM optical disk such as disk 32 inserted into the optical disk drive 18. Note that these instructions may be obtained already encoded on the program storage medium or, in some embodiments, downloaded to the same from the World Wide Web. Specification, p. 12.

This particular embodiment also employs a form-type, graphical user interface frequently called a “wizard” to interface with the installer. While the interaction between with the installer is reflected in Figures 4A-4C, references to such activities are sometimes omitted from the following discussion for the sake of clarity and brevity. Accordingly, while perhaps preferable in some embodiments, the use of a wizard is not necessary to the practice of the invention. Some embodiments may actually be entirely automated and thus eliminate the need for intermediate installer interaction. Specification, pp. 12-13.

Referring now to Figure 4A, the computer 12 in this particular embodiment first presents an introduction screen to the installer requesting the selection of one of four installation options.

The four options in this particular embodiment are: (1) install Windows® 95 while keeping the existing configuration; (2) install Windows® 95 keeping the existing internal configuration files only, (3) install Windows® 95 after removing all EXISTING Windows® files; and (4) install Windows® 95 after formatting the hard disk 20. Options (2)-(4) require intermediate preparation. Depending on whether options (2)-(4) are selected, the computer: removes the internal configuration files previously installed in C:\WINDOWS; removes all the Windows® 95 files, including the internal configuration file, and then deletes the C:\WINDOWS directory; or formats the hard disk, respectively. As will be known to the art, Windows® configuration files end with the extension .INI, such that the configuration files can be readily deleted by deleting C:\WINDOWS\\*.INI. Specification, p. 13.

If option (1) is selected and, if options (2)-(4) are selected, after any intermediate preparation, the computer 12 copies all necessary files from the D:\OSI directory of the portable program storage medium to a newly created C:\WIN95 directory on the hard disk 20. "Necessary files" in this case are those files of all types necessary to install Windows® 95. The computer 12 then reads the configuration information from the external configuration file C:\WIN95\OSI.INI containing the configuration information for Windows® 95. Note that OSI.INI is stored in C:\WIN95 at this point. The computer 12 then prompts the installer to select the display, sound, modem, and SCSI device type. This may be done by asking the installer to select from a list of presented options or the installer may simply enter the information when prompted. Specification, pp. 13-14.

Utilizing the information from the external configuration file OSI.INI and from the device selection, the computer 12 then generates two temporary batch files (OSI1.BAT and OSI6.BAT) and a registry file (OSIREG.INI) as set forth in Figure 4A. The registry file

(OSIREG.INI) will subsequently be used to write correct configuration information to the registry file for the Windows® 95 operating system. Four other temporary batch files (OSI2.BAT, OSI3.BAT, OSI4.BAT, and OSI5.BAT), whose functions are later discussed, are previously generated and stored on the portable program storage medium (not shown). Batch files are simply files containing a series of commands for a computer to execute. Thus, the contents of the temporary batch files will depend on their functions disclosed below and may vary depending on the particular embodiment implemented. The computer 12 then begins installing the operating system. Specification, p. 14.

The computer 12 starts the installation by executing the first temporary batch file OSI1.BAT. The Windows® 95 files are copied from the CD-ROM installation disk(s) to the C:\WIN95 directory on the hard disk 20. Next, the drivers for each device are copied from a respective installation disk to a C:\WIN95\INF subdirectory on the hard disk 20. When the Windows® 95 files and device drivers have been completely copied, the first temporary batch file is copied over the AUTOEXEC.BAT file on the hard disk 20 and renamed AUTOEXEC.BAT. The AUTOEXEC.BAT file is the first file executed by the computer 12 when it is turned on or rebooted. Thus, the next time the computer 12 is booted, it will execute the first temporary batch file OSI1.BAT. Specification, pp. 14-15.

The computer 12 then begins the Windows® setup using the parameters that force hardware detection, bypass the license screen, bypass ScanDisk, bypass the disk space check. ONI.INF is specified as the information setup file for the setup. The information setup file contains some of the information necessary to install the driver for each device previously selected. Such information includes the memory location reserved for the device and any necessary registry changes. The installer inputs the appropriate Windows® 95 product



identification number and the computer is rebooted. Upon reboot, the computer 12 executes AUTOEXEC.BAT as it always does, and thus executes the first temporary batch file (OSI1.BAT) that was generated earlier and copied over the original AUTOEXEC.BAT. Specification, p. 15.

The AUTOEXEC.BAT then calls the second temporary batch file (OSI2.BAT) from the C:\WIN95 directory of the hard disk 20 and executes it. OSI2.BAT copies the information files (\*.INF) from the C:\WIN95\INF subdirectory to the C:\WINDOWS\INF subdirectory. After copying, the C:\WINDOWS\INF subdirectory provides the information needed by Windows® 95 to install the newer drivers Windows® 95 otherwise would omit or install erroneously. OSI2.BAT then copies the third temporary file (OSI3.BAT) to AUTOEXEC.BAT and setup proceeds automatically from the information in OSI.INF. Then, the installer selects their time zone and Setup reboots the computer 12. Specification, pp. 15-16.

On reboot, the computer 12 executes AUTOEXEC.BAT, which is now what formerly was the third temporary batch file OSI3.BAT. AUTOEXEC.BAT imports the correct information into the registry to run when Windows® 95 starts. AUTOEXEC.BAT then runs the fourth temporary batch file (OSI4.BAT) from the C:\WIN95 directory, which copies the fifth temporary batch file (OSI5.BAT) over AUTOEXEC.BAT. OSI4.BAT then proceeds by starting Windows® 95 by starting OSIREG.EXE in the C:\WIN95 directory. OSIREG.EXE then reads the necessary information from OSIREG.INI, removes erroneous entries in the registry as defined by OSIREG.INI. If an internal modem is configured, the external COM2 I/O port is disabled in the registry. The computer 12 is then rebooted. Specification, p. 16.

On reboot, AUTOEXEC.BAT, which was formerly OSI5.BAT, begins running by calling the sixth temporary batch file (OSI6.BAT). OSI6.BAT then imports any registry files for the

devices into the registry to fix any errors in the way Windows® 95 detected new devices in the error. This fix is known as a "patch." OSI6.BAT then deleted AUTOEXEC.BAT and Windows® 95 is installed. If the installer selected option 4, then installation of applications programs can proceed in the manner known to the art. Specification, p. 16.

## VI. ISSUES

- A. **Can references that do not teach or suggest providing a second configuration file that is external to an operating system package render claims 34-43 obvious?**
- B. **Can references that do not teach or suggest instructions to cause a computer to generate an external configuration file that is external to an operating system package and install a driver for a particular device of the computer based on the information in the external configuration file render claims 44-51 obvious?**

## VII. GROUPING OF THE CLAIMS

Claims 34-43 can be grouped together; and claims 44-51 can be grouped together.

## VIII. ARGUMENT

All claims should be allowed over the cited references for the reasons set forth below.

- A. **Can references that do not teach or suggest providing a second configuration file that is external to an operating system package render claims 34-43 obvious?**

The method of claim 34 includes providing an operating system package that includes a first configuration file. This first configuration file includes information that is used by the operating system package to install first drivers for a first set of devices on a computer. The method also includes providing a second configuration file that is external to the operating system package. This second configuration file includes information to direct the installation of a second driver for a second device.

Many possible advantages may flow from embodiments of the invention. For example, because the external, second configuration file is not part of the operating system package, it

may be frequently updated or modified as new devices are released or errors are identified in an internal configuration file of the operating system package. Other and different advantages are possible in the various embodiments of the invention.

The Examiner combines alleged admitted art in the Background section of the present application in view of Lipe (U.S. Patent No. 5,748,980) in rejecting claims 34-43 under 35 U.S.C. § 103(a). In particular, the Examiner relies on Lipe to teach the act of providing a second configuration file that is external to an operating system package. In this reliance, the Examiner points to a configuration manager 158 of Lipe. However, this configuration manager 158 is part of an operating system 10 that is described by Lipe and is not external to the operating system 10. In this manner, Lipe specifically states in col. 26, ll. 26-30, “the operating system 10 comprises...a configuration manager 158.” Thus, the configuration manager 158 is part of the operating system 10. Furthermore, there is no language in Lipe that contradicts the above-recited quotation from Lipe.

In response to pointing out the above-recited quotation from Lipe, the Examiner states that “Lipe’s configure manager 158 is separate from the OS, and one skilled in the art would readily distinguish the config. 158 module from the files/utilities that are inextricable part of the OS.” However, the Examiner fails to provide support for such an interpretation, as this interpretation of Lipe is in stark contrast to the explicit language of Lipe. Therefore, for at least the reason that Lipe not only fails to teach missing claim limitations but in fact teaches the opposite of these claim limitations, the Examiner has failed to establish a *prima facie* case of obviousness. M.P.E.P. § 2143.

Thus, the rejections of claims 34-43 should be reversed.

**B. Can references that do not teach or suggest instructions to cause a computer to generate an external configuration file that is external to an operating system package and install a driver for a particular device of the computer based on the information in the external configuration file render claims 44-51 obvious?**

The article of claim 44 includes at least computer readable storage medium that stores instructions to cause a computer to generate an external configuration file that is external to an operating system package. The instructions also cause the computer to install a driver for a particular device of the computer based on the information in the external configuration file.

The Examiner applies the alleged admitted art in the Background section of the present application and Lipe in rejecting claims 44-51 under 35 U.S.C. § 103(a). In particular, the Examiner relies on Lipe to teach instructions to cause a computer to provide a second configuration file that is external to an operating system package and install a second driver for a particular device of the computer based on the information in the second configuration file. However, as noted above in connection with Issue A, Lipe teaches the opposite of the alleged teaching on which the Examiner relies. In this manner, Lipe teaches a configuration file that is part of an operating system, not a configuration file that is external to an operating system package. Therefore, for at least this reason, the Examiner has failed to establish a *prima facie* case of obviousness.

Thus, the rejections of claims 44-51 should be reversed.

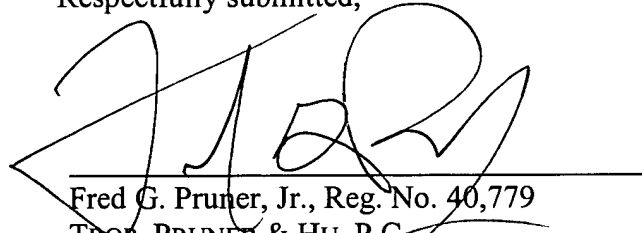
IX. CONCLUSION

The Applicant requests that each of the final rejections be reversed and that the claims subject to this appeal be allowed to issue.

Respectfully submitted,

Date: \_\_\_\_\_

10/02/01



Fred G. Pruner, Jr., Reg. No. 40,779  
TROP, PRUNER & HU, P.C.  
8554 Katy Freeway, Suite 100  
Houston, Texas 77024-1805  
713/468-8880 [Phone]  
713/468-8883 [Facsimile]

## APPENDIX OF CLAIMS

The claims on appeal are:

34. A method comprising:

providing an operating system package that includes a first configuration file, the first configuration file including information used by the operating system package to install first drivers for a first set of devices on a computer;

providing a second configuration file external to the operating system package, the second configuration file including information to direct the installation of a second driver for a second device;

installing the second driver on the computer based on the information in the second configuration file; and

installing the operating system package on the computer based on the information in the second configuration file.

35. The method of claim 34, further comprising:

executing a setup program of the operating system package to install the second driver.

36. The method of claim 34, wherein the second device is not included in the first set of devices.

37. The method of claim 34, further comprising installing applications after the second driver and the operating system are installed.

38. The method of claim 34, further comprising determining one installation procedure from among a plurality of options.

39. The method of claim 34, wherein the second device is identified dynamically.

40. The method of claim 39, wherein the dynamic identification is at least one of automated and user-driven.

41. The method of claim 34, wherein the second device is at least one of an ISA, PCI, SCSI, and an IDE device.

42. The method of claim 34, wherein the second device is at least one of a display device, a sound device, a modem, and a controller.

43. The method of claim 34, wherein installing the operating system includes patching errors in the first configuration file with information in the second configuration file.

44. An article comprising at least one computer readable storage medium storing instructions to cause a computer to:

generate an external configuration file that is external to an operating system package, the external configuration file including information to direct the installation of a driver for a particular device and the operating system package including an internal configuration file used by the operating system package to install other drivers for other devices on the computer;

install the driver for the particular device of the computer based on the information in the external configuration file; and

install the operating system package on the computer based on the information in the external configuration file.

45. The article of claim 44, wherein said at least one storage medium stores instructions to cause the computer to install applications after the particular driver and the operating system package are installed.

46. The article of claim 44, wherein said at least one storage medium stores instructions to cause the computer to determine one installation procedure from among a plurality of options.

47. The article of claim 44, wherein said at least one storage medium stores instructions to cause the computer to dynamically identify the particular device.

48. The article of claim 44, wherein the particular device is at least one of an ISA, PCI, SCSI, and an IDE device.

49. The article of claim 44, wherein the particular device is at least one of a display device, a sound device, a modem, and a controller.

50. The article of claim 44, wherein said at least one storage medium stores instructions to cause the computer to install the operating system by a technique that includes patching errors in an OS configuration file with information in the external configuration file.

51. The article of claim 44, wherein said at least one storage medium stores instructions to cause the computer to install the driver for the second device and the operating system package contemporaneously.